IN THE CLAIMS:

A complete listing of the claims is set forth below. Please amend the claims as follows:

1. **(Currently Amended)** A component-based distributed software system, comprising:

a first container comprising:

at least one server component capable of having a client-server relationship with one or more client components;

one or more ~~supporting one or more~~ server objects having associated data and capable of being supported by the at least one server object; and

one or more client components which are local to the at least one server component; ~~having associated data, the server component being within a first container;~~ and

a second container comprising at least one proxy component, one or more proxy objects capable of being supported by the at least one server object, ~~at least one client component that is within a second container,~~ and one or more client components capable of having a client-server relationship with one or more server components, the one or more client components are remote and distributed from the at least one server component, and operable to:

access data associated with one or more of the server objects such that whether the server component is local to or remote from the client component is substantially transparent to the client component;

if the server component ~~first container~~ is local to the client component, ~~second container,~~ in order to access server object data, execute data access operations optimized for local communications; and if the server component ~~first container~~ is remote from the client component, ~~second container,~~ in order to access server object data, access at least one proxy component that is:

within the second container;

supporting one or more proxy objects each providing a local version of a corresponding server object: and

operable to:

provide the client component with access to data associated with a proxy object in response to the client component requesting data associated with the corresponding server object;

execute data access operations optimized for remote communications to access data associated with the corresponding server object; and

substantially immediately reflect all changes to data associated with the proxy objects back to data associated with the corresponding server objects.

2. **(Previously Presented)** The system of Claim 1, wherein the client component is allowed to use the same operations to access server object data whether the client component is local to or remote from the server component.

3. **(Original)** The system of Claim 1, wherein the client component is operable to access server object data without determining whether the client component is local to or remote from the server component.

4. **(Original)** The system of Claim 1, wherein the client component is coded as if the client component will always be remote from any associated server component and all communications to such a server component will be remote communications.

5. **(Original)** The system of claim 1, wherein the client component has been developed using templatized code appropriate for multiple client components of the server component, such that local and remote client-server interface transparency is preserved across all such client components and repetitive code generation has been minimized in developing such client components.

6. **(Cancelled)**

7. **(Cancelled)**

8. **(Previously Presented)** The system of Claim 1, wherein the proxy component is operable to perform management tasks relating to the proxy objects.

9. **(Previously Presented)** The system of Claim 1, wherein the proxy component is a generic component customized by a developer of the server component.

10. **(Previously Presented)** The system of Claim 1, wherein the proxy component and the server component are operable to cooperate to reconcile proxy object data with server object data, using one or more operations, in a manner consistent with local and remote client-server interface transparency.

11. **(Currently Amended)** A component-based distributed software system, comprising:

a first container comprising:

at least one server component capable of having a client-server relationship with one or more client components;

one or more ~~supporting one or more~~ server objects having associated data and capable of being supported by the at least one server object; and

one or more client components which are local to the at least one server component; ~~having associated data, the server component being within a first container;~~ and

a second container comprising at least one proxy component, one or more proxy objects capable of being supported by the at least one server object, ~~at least one client component that is within a second container,~~ and one or more client components capable of having a client-server relationship with one or more server components, the one or more client components are remote and distributed from the at least one server component, and operable to:

access data associated with one or more of the server objects according to a scheme allowing the client component to use the same operations to access server object data whether the client component is local to or remote from the server component;

if the server component ~~first container~~ is local to the s client component, ~~second container,~~ in order to access server object data, execute data access operations optimized for local communications; and

if the server component ~~first container~~ is remote from the client component, ~~second container,~~ in order to access server object data, access at least one proxy component that is:

within the second container;

supporting one or more proxy objects each providing a local version of a corresponding server object; and

operable to:

provide the client component with access to data associated with a proxy object in response to the client component requesting data associated with the corresponding server object;

execute data access operations optimized for remote communications to access data associated with the corresponding server object; and

substantially immediately reflect all changes to data associated with the proxy objects back to data associated with the corresponding server objects.

12. **(Currently Amended)** A component-based distributed software system, comprising:

a first container comprising:

at least one server component capable of having a client-server relationship with one or more client components;

one or more ~~supporting one or more~~ server objects having associated data and capable of being supported by the at least one server object; and

one or more client components which are local to the at least one server component; ~~having associated data, the server component being within a first container;~~ and

a second container comprising at least one proxy component, one or more proxy objects capable of being supported by the at least one server object, ~~at least one client component that is~~ and one or more client components capable of having a client-server relationship with one or more server components, the one or more client components are remote and distributed from the at least one server component, ~~the client component being within a second container~~ and operable to:

access data associated with one or more of the server objects, without determining whether the client component is local to or remote from the server component such that whether the server component is local to or remote from the client component is substantially transparent to the client component and such that the client component is allowed to use the same operations to access server object data whether the client component is local to or remote from the server component;

if the server component ~~first container~~ is local to the client component, ~~second container,~~ in order to access server object data, execute data access operations optimized for local communications; and

if the server component ~~first container~~ is remote from the client component, ~~second container,~~ in order to access server object data, access at least one proxy component supporting one or more proxy objects each providing a local version of a corresponding server object, the proxy component being within the second container and operable to:

provide the client component with access to data associated with a proxy object in response to the client component requesting data associated with the corresponding server object;

execute data access operations optimized for remote communications to access data associated with the corresponding server object; and

substantially immediately reflect all changes to data associated with the proxy objects back to data associated with the corresponding server objects.

13. **(Currently Amended)** A client ~~component~~ component, within a first container, that is distributed from a server ~~component~~ component, within a second container, supporting one or more server objects having associated data and operable to:

access data associated with one or more of the server objects such that whether the server component is local to or remote from the client component is substantially transparent to the client component;

if the server component ~~first container~~ is local to the client component, ~~second container,~~ in order to access server object data, execute data access operations optimized for local communications; and

if the server component ~~first container~~ is remote from the client component, ~~second container,~~ in order to access server object data, access at least one proxy component that is:

within the second container;

supporting one or more proxy objects each providing a local version of a corresponding server object; and

operable to:

provide the client component with access to data associated with a proxy object in response to the client component requesting data associated with the corresponding server object;

execute data access operations optimized for remote communications to access data associated with the corresponding server object; and

substantially immediately reflect all changes to data associated with the proxy objects back to data associated with the corresponding server objects.

14. **(Previously Presented)** A proxy component that:

is within a first container that also contains a client component and is remote from a second container containing a server component which supports one or more server objects having associated data, the client component being distributed from the server component and operable to execute data access operations optimized for local communications to access data associated with a corresponding server object;

supports one or more proxy objects each providing a local version of a corresponding server object;

is operable to provide the client component with access to data associated with a proxy object in response to the client component requesting data associated with the corresponding server object, such that whether the server component is local to or remote from the client component is substantially transparent to the client component;

is operable to execute data access operations optimized for remote communications to access data associated with the corresponding server object; and

substantially immediately reflect all changes to data associated with the proxy objects back to data associated with the corresponding server objects.

15. **(Currently Amended)** A method of providing data access in a component-based distributed software system, comprising:

receiving a request from a client ~~component~~ component, within a first container, for data that is associated with a server object of a server component;

if the client component is local to the server component, allowing the client component to directly access the requested server object data, the client component operable to execute data access operations optimized for local communications to access server object data;

if the client component is remote from the server component, using a proxy component to provide the client component with local access to proxy object data corresponding to the requested server object data, the proxy component supporting one or more proxy objects each being a local copy of a corresponding server object, the proxy component operable to execute data access operations optimized for remote communications to access data associated with the corresponding server object; and

substantially immediately reflect all changes to data associated with the proxy objects back to data associated with the corresponding server objects;

wherein whether the server component is local to or remote from the client component is substantially transparent to the client component.

16. **(Original)** The method of Claim 15, further comprising allowing the client component to use the same operations to access server object data whether the client component is local to or remote from the server component.

17. **(Original)** The method of Claim 15, further comprising allowing the client component to access server object data without determining whether the client component is local to or remote from the server component.

18. **(Original)** The method of Claim 15, wherein the client component is coded as if the client component will always be remote from any associated server component and all communications to such a server component will be remote communications.

19. **(Original)** The method of Claim 15, wherein the client component has been developed using templatized code appropriate for multiple client components of the server component, such that local and remote client-server interface transparency is preserved across all such client components and repetitive code generation has been minimized in developing such client components.

20. **(Cancelled)**

21. **(Cancelled)**

22. **(Previously Presented)** The method of Claim 15, wherein the proxy component performs management tasks relating to the proxy objects.

23. **(Previously Presented)** The method of Claim 15, wherein the proxy component is a generic component customized by a developer of the server component.

24. **(Previously Presented)** The method of Claim 15, wherein the proxy component and the server component cooperate to reconcile proxy object data with server object data, using one or more operations, in a manner consistent with local and remote client-server interface transparency.

25. **(Currently Amended)** A method of providing data access in a component-based distributed software system, comprising:

receiving a request from a client ~~component~~ component, within a first container, for data that is associated with a server object of a server component;

if the client component is local to the server component, allowing the client component to directly access the requested server object data, the client component operable to execute data access operations optimized for local communications to access server object;

if the client component is remote from the server component, using a proxy component to provide the client component with local access to proxy object data corresponding to the requested server object data, the proxy component supporting one or more proxy objects each being a local copy of a corresponding server object, the proxy component operable to execute data access operations optimized for remote communications to access data associated with the corresponding server object; and

substantially immediately reflect all changes to data associated with the proxy objects back to data associated with the corresponding server objects;

the client component using the same operations to access server object data whether the client component is local to or remote from the server component.

26. **(Previously Presented)** A method of providing data access in a component-based distributed software system, comprising:

receiving a request from a client component for data that is associated with a server object of a server component distributed from the client component, the server component being within a first container and the client component being within a second container, the client component operable to execute data access operations optimized for local communications to access server object data;

if the client component is local to the server component, allowing the client component to directly access the requested server object data;

if the client computer is remote from the server component, using a proxy component within the second container to provide the client component with local access to proxy object data corresponding to the requested server object data, the proxy component supporting one or more proxy objects each being a local version of a corresponding server object, the proxy component operable to execute data access operations optimized for remote communications to access data associated with the corresponding server object; and

substantially immediately reflect all changes to data associated with the proxy objects back to data associated with the corresponding server objects;

wherein whether the server component is local to or remote from the client component is substantially transparent to the client component, the client component being able to use the same data access operations whether the client component is local to or remote from the server component.

27. **(Currently Amended)** A method of accessing data in a component-based distributed software system using a client component, comprising:

at the client component, within a first container, accessing data associated with one or more server objects of a server component that is distributed from the client component, each server object having associated data, the client component accessing the server object data such that whether the server component is local to or remote from the client component is substantially transparent to the client component;

if the client component is local to the server component, allowing the client component to directly access the requested server object data, the client component operable to execute data access operations optimized for local communications to access data associated with one or more server objects;

if the client component is remote from the server component, using a proxy component to provide the client component with local access to proxy object data corresponding to the requested server object data, the proxy component supporting one or more proxy objects each being a local copy of a corresponding server object, the proxy component operable to execute data access operations optimized for remote communications to access the data associated with the corresponding server object; and

substantially immediately reflect all changes to data associated with the proxy objects back to data associated with the corresponding server objects.

28. **(Previously Presented)** A method of providing data access in a component-based distributed software system using a proxy component, the proxy component operable to execute data access operations optimized for local communications, the proxy component being within a first container that also contains a client component and is remote from a second container containing a server component supporting one or more server objects having associated data, the client component being distributed from the server component and operable to execute data access operations optimized for local communications the method comprising:

supporting one or more proxy objects each providing a local version of a corresponding server object;

providing the client component with access to data associated with a proxy object in response to the client component requesting data associated with the corresponding server object, such that whether the server component is local to or remote from the client component is substantially transparent to the client component and such that data access operations optimized for remote communications are performed when the client component is remote from the server component and data access operations optimized for local communications are performed when the client component is local to the server component; and

substantially immediately reflect all changes to data associated with the proxy objects back to data associated with the corresponding server objects.